



MLTE: A Unicode Text Engine

Session 208





MLTE: A Unicode Text Engine

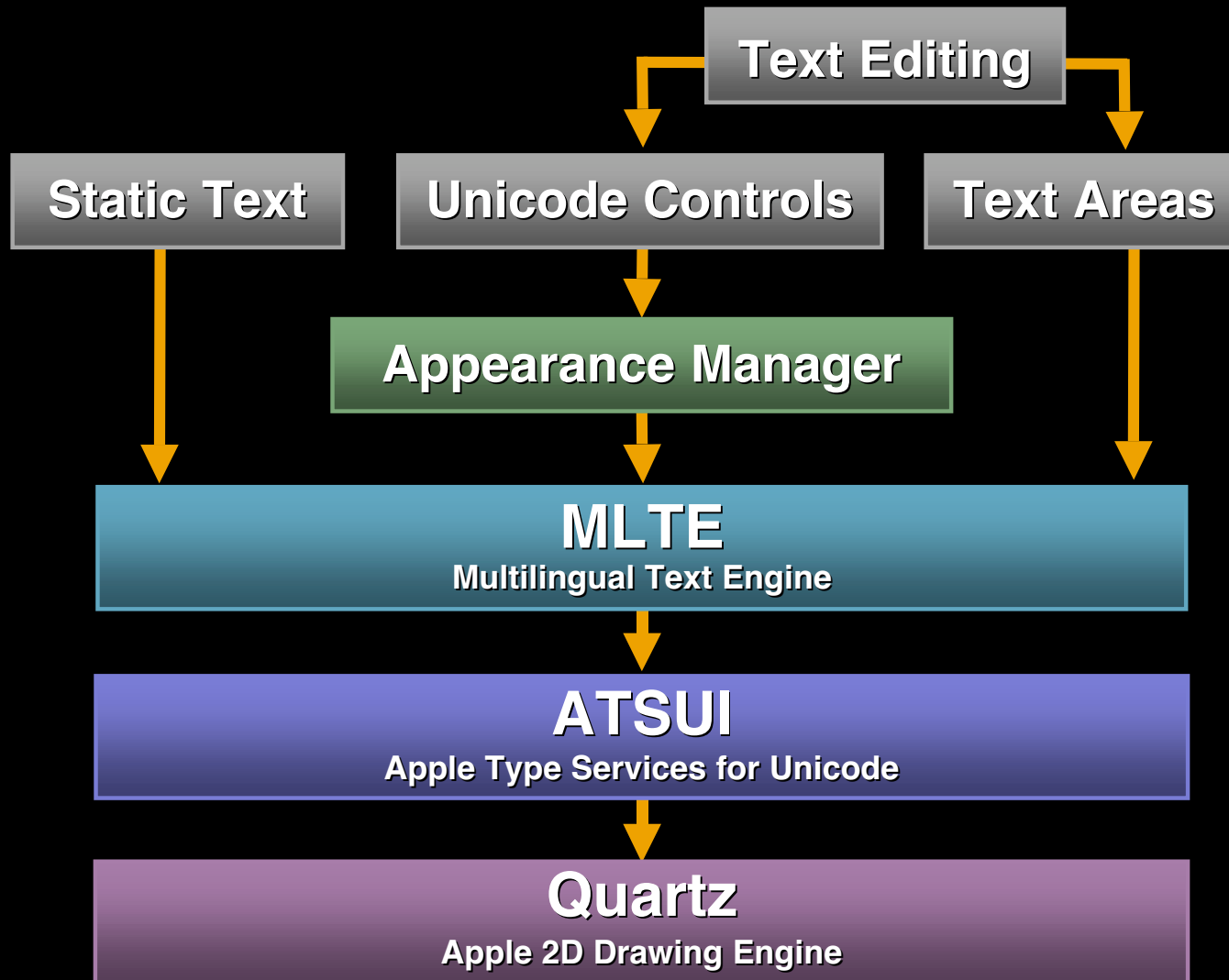
Xavier Legros
Mac OS X Evangelist

Today's Presentation

- MLTE overview
- MLTE concepts
- Basic tasks
- Advanced tasks
- New tasks—What you can do with Jaguar



Text on Carbon





MLTE Overview

Rick Fabrick
Software Engineer

MLTE Overview

- Carbon Unicode text engine API
- Basic operations are handled automatically
 - Multilingual text input and layout
 - Quartz imaging
 - Drag and drop
 - Scrollbar handling
 - Selection
 - Multimedia support (pictures, movies, and sounds)



MLTE Overview

- Additional operations are supported by a simple set of API
 - Document manipulation (Open, Save, Revert, Print)
 - Copy and paste
 - Multiple undo and redo
 - Font menu
 - Find
 - Password mode
 - Cursor updates
- Support for advanced typography features
- No 32K text limit!



New MLTE Features in Jaguar

- Automatic URL detection and browser launch
- Support for monostyled text
- Functions to support scrolling with controls created by client
- Function to clear the undo and redo stack
- Support for mouse scroll wheel

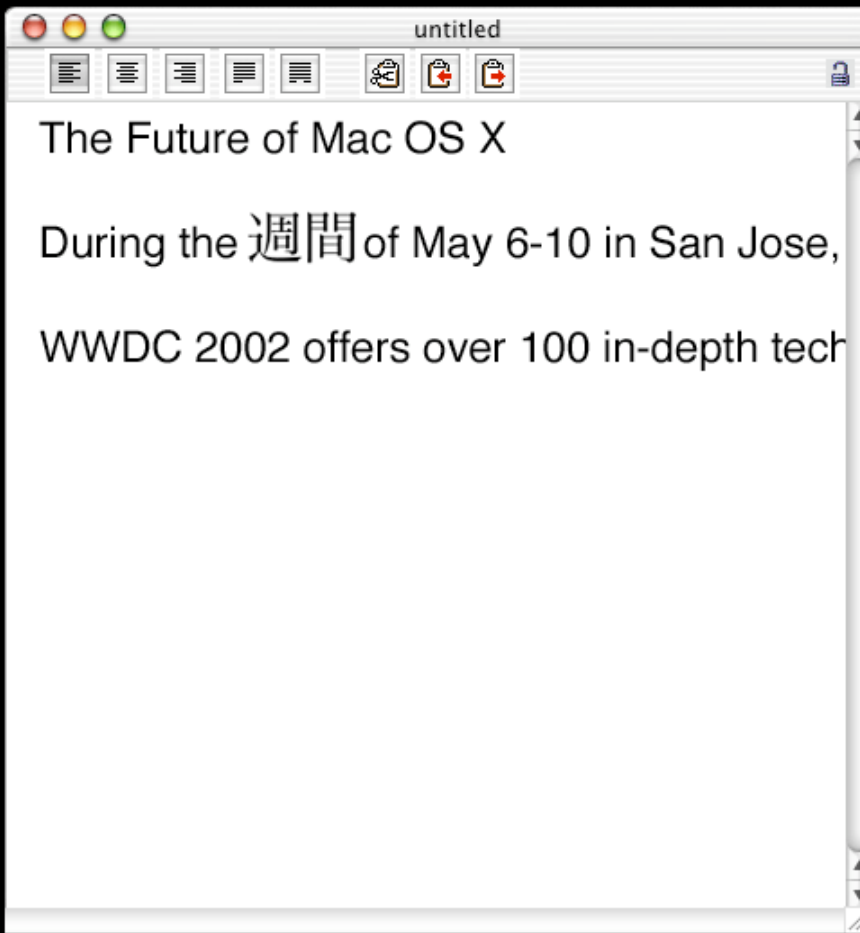


MLTE Concepts

- `TXNObject` is an opaque structure
- An object is either a `window` or a `pane` object
- Object controls define object-wide behavior
- Style attributes define appearance of text ranges
- Carbon events are installed to automatically handle basic user interactions



TXNSetTXNObjectControls



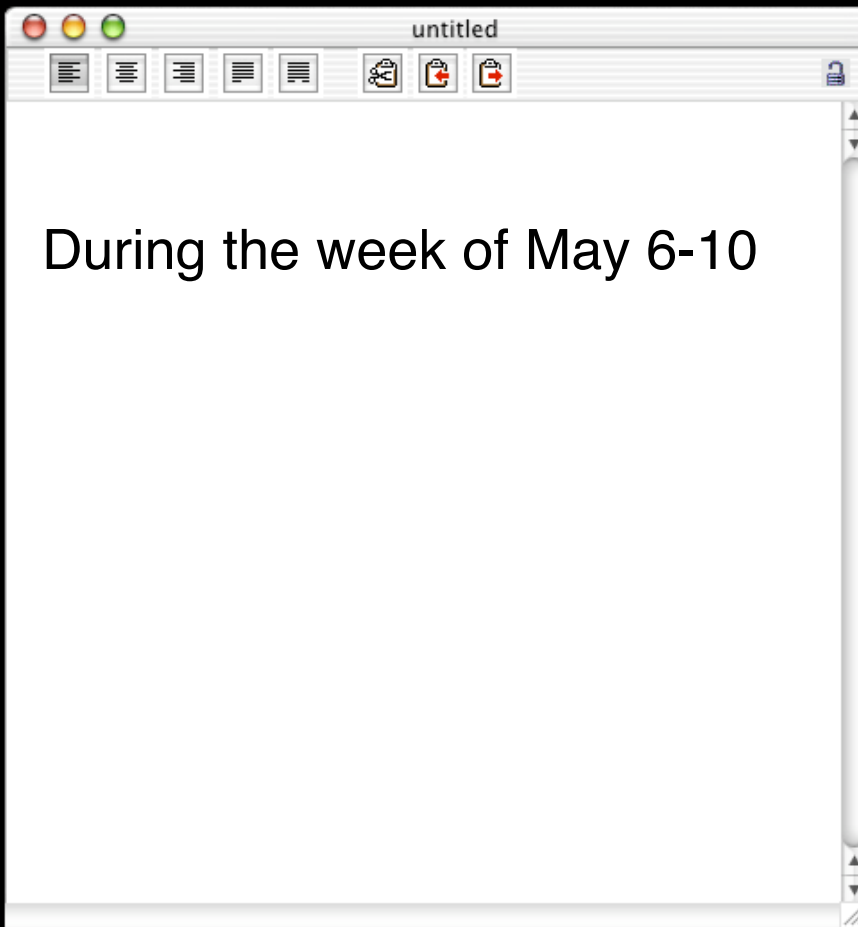
kTXNDoFontSubstitution

kTXNJustificationTag

kTXNWordWrapStateTag



TXNSetTypeAttributes



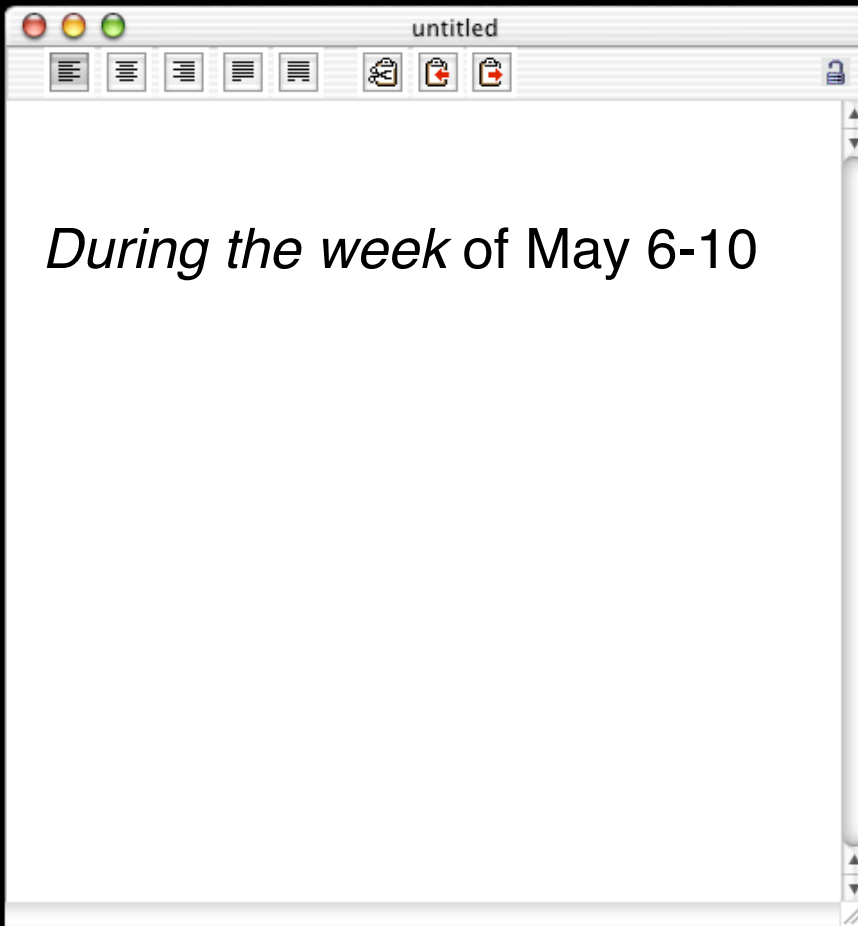
Style

During the week of May 6-10

Plain



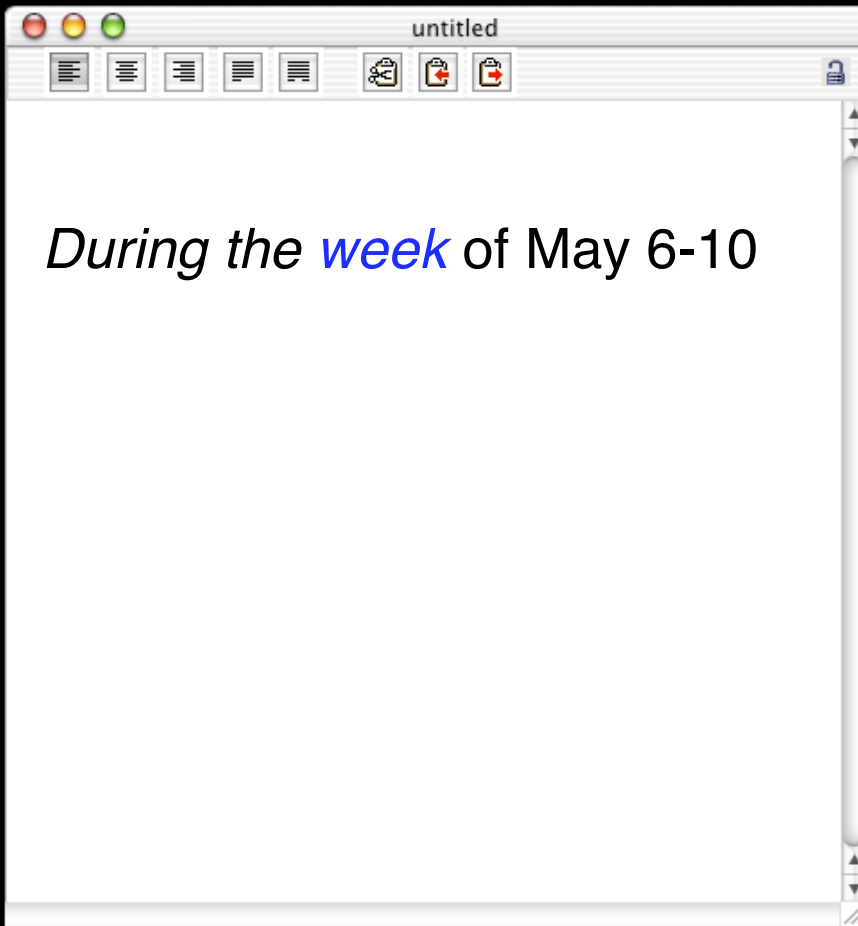
TXNSetTypeAttributes



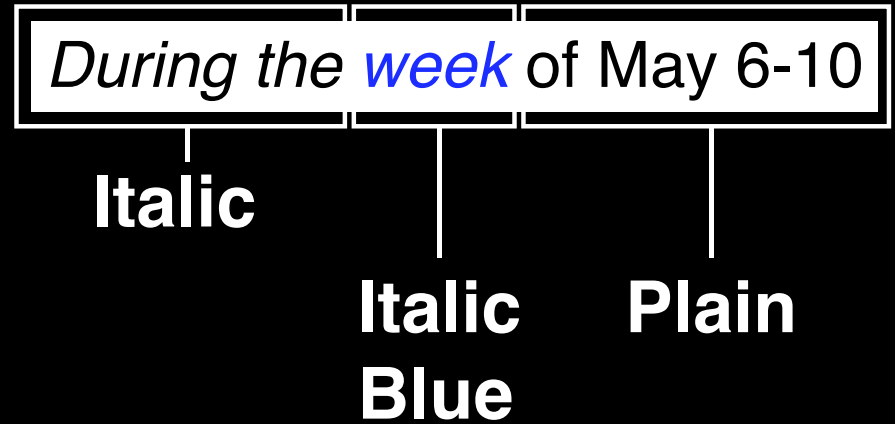
Style



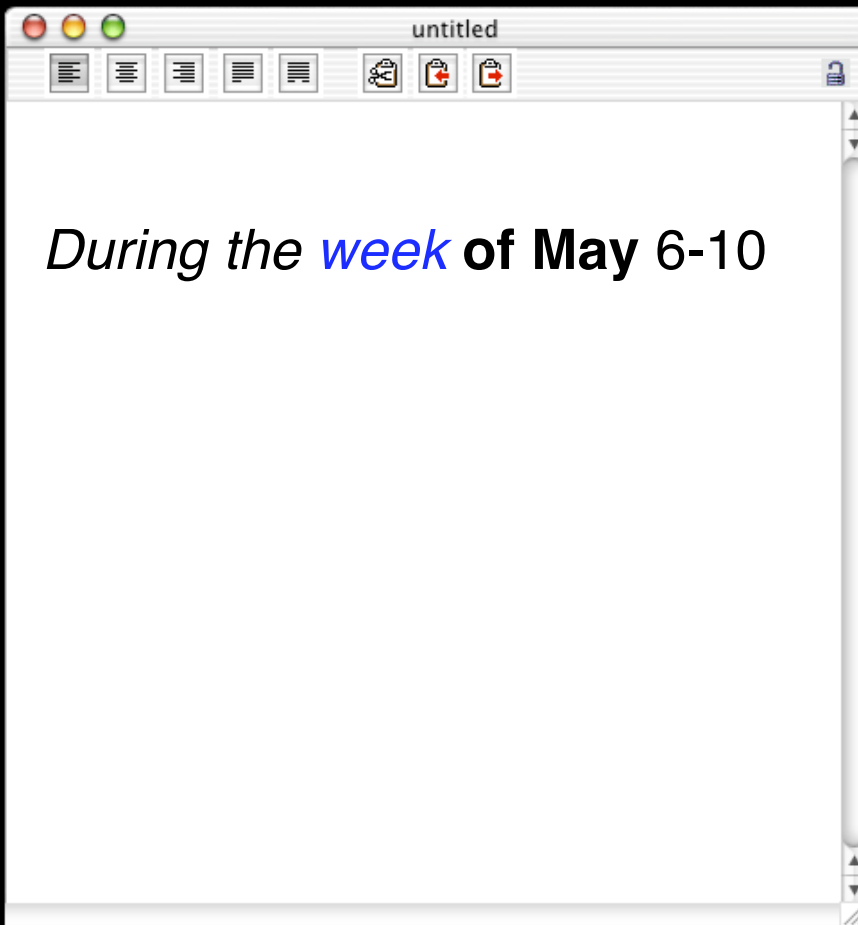
TXNSetTypeAttributes



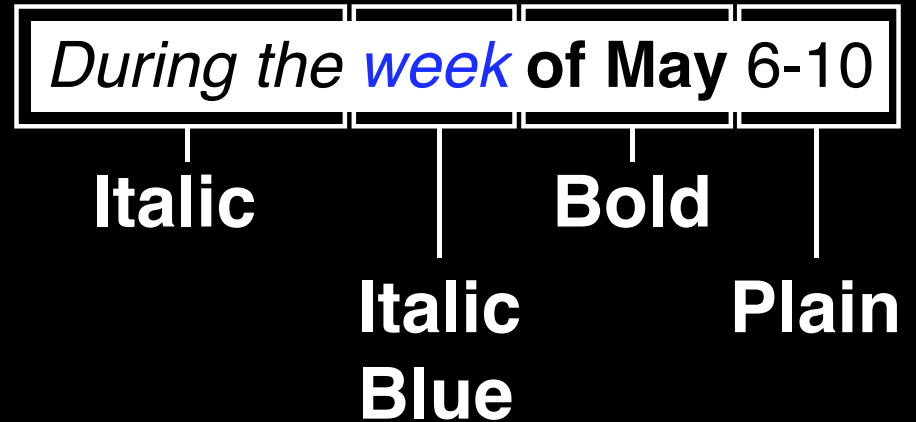
Style



TXNSetTypeAttributes



Style



Carbon Events Installed Automatically

- Handlers for text input from keyboards and input methods
- Handler for mouse-click operations such as selection and scrolling
- Handlers for window events such as activate, deactivate, resize, and update
 - Only installed for window objects



Today's Presentation

- MLTE overview
- MLTE concepts
- Basic tasks
 - Working with TXNObject
 - Using Unicode Control
 - Drawing Static Text
- Advanced tasks
- New tasks—What can you do with Jaguar



Working With TXNObject

- Creating an MLTE Object

```
OSStatus TXNNewObject ( const FSSpec * iFileSpec, WindowRef iWindow, const Rect * iFrame,  
TXNFrameOptions iFrameOptions, TXNFrameType iFrameType,  
TXNFileType iFileType,  
TXNPermanentTextEncodingType iPermanentEncoding,  
TXNObject * oTXNObject, TXNFrameID * oTXNFrameID,  
TXNObjectRefcon iRefCon );
```

- Adding data to the object

```
OSStatus TXNSetData ( TXNObject iTXNObject, TXNDataType iDataType,  
const void * iDataPtr, ByteCount iDataSize,  
TXNOffset iStartOffset, TXNOffset iEndOffset );
```

- Changing default behavior of entire object

```
OSStatus TXNSetTXNObjectControls ( TXNObject iTXNObject, Boolean iClearAll,  
ItemCount iControlCount,  
const TXNControlTag iControlTags[],  
const TXNControlData iControlData[] );
```



Working With TXNObject

- Change style of text runs

```
OSStatus TXNSetTypeAttributes ( TXNObject iTXNObject, ItemCount iAttrCount,  
                                const TXNTypeAttributes iAttributes[],  
                                TXNOffset iStartOffset, TXNOffset iEndOffset );
```

- Saving the object to a file

```
OSStatus TXNSave ( TXNObject iTXNObject, TXNFileType iType, OSType iResType,  
                   TXNPermanentTextEncodingType iPermanentEncoding,  
                   const FSSpec * iFileSpecification, SInt16 iDataReference,  
                   SInt16 iResourceReference );
```

- Printing the object to a printer

```
OSStatus TXNPageSetup ( TXNObject iTXNObject );  
OSStatus TXNPrint ( TXNObject iTXNObject );
```



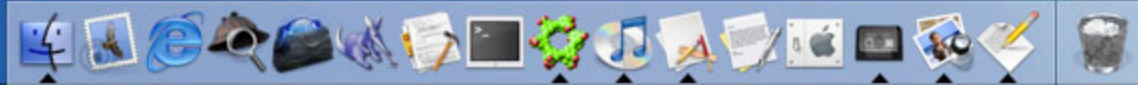
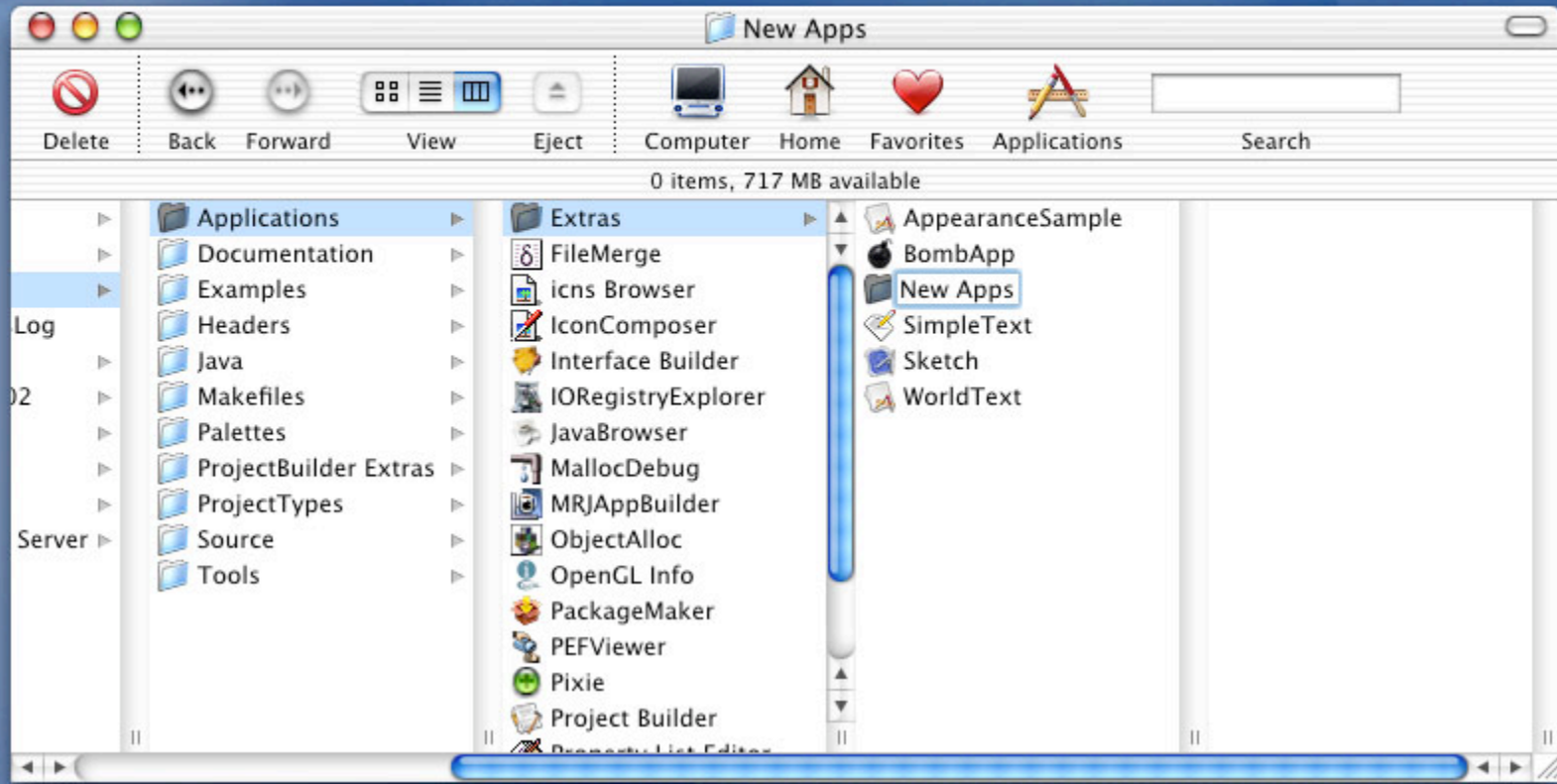
Unicode Edit Text Control

- Inherits most MLTE features
- Conforms to the Control Manager interface
 - EventTargetRef for Carbon event handlers
- Unicode replacement for EditText control
- Creating the control

```
OSStatus CreateEditUnicodeTextControl ( WindowRef window, const Rect * boundsRect,  
                                       CFStringRef text, Boolean isPassword,  
                                       const ControlFontStyleRec * style,  
                                       ControlRef * outControl );
```

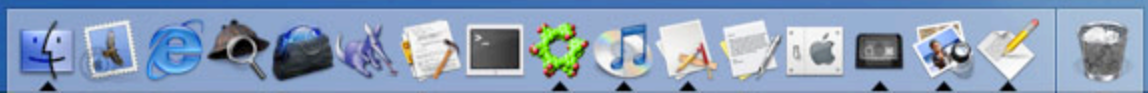
- Controls support single line option (new in Jaguar)
 - Use `kControlEditTextSingleLineTag` to toggle on and off





Key Caps

これは MLTE خانة



WWDC 鼻 (notes)

Save WWDC 鼻 (notes)

Where: Desktop

Cancel Save





iTunes

A Momentary Lapse of Reason
Elapsed Time: 0:01

Search Browse

Source	Song	Time	Artist	Album
Library	<input checked="" type="checkbox"/> Signs of Life	4:24	Pink Floyd	A Momentary Lapse of Reason
Radio Tuner	<input checked="" type="checkbox"/> Learning to Fly	4:53	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> The Dogs of War	6:10	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> One Slip	5:04	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> On the Turning Away	5:41	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> Yet Another Movie	7:28	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> New Machine Part One	1:46	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> Terminal Frost	6:17	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> A New Machine Part 2	0:38	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> Sorrow	8:48	Pink Floyd	A Momentary Lapse of Reason
	<input checked="" type="checkbox"/> Speak to Me/Breathe in the Air	4:00	Pink Floyd	Dark Side of the Moon
	<input checked="" type="checkbox"/> On the Run	3:33	Pink Floyd	Dark Side of the Moon

1354 songs, 4:00:50:45 total time, 6.52 GB

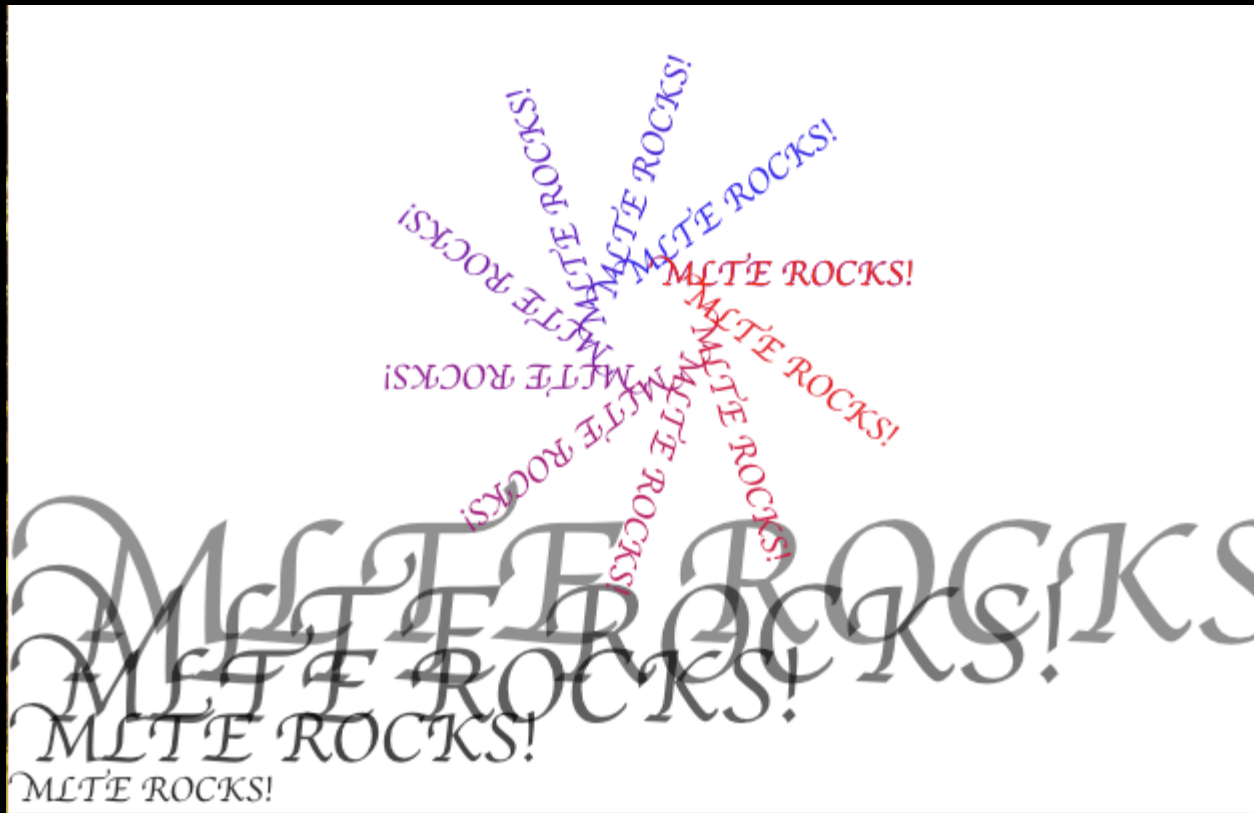


Drawing Static Text

- In most cases use DrawThemeTextBox API
- Use TXNDrawUnicodeTextBox or TXNDrawCFStringTextBox if you need to:
 - Pass your own CG context
 - Have ATSUI styles
 - Draw rotated text
 - Draw vertical text
- Use ATSUI if you need even more control



TXNDrawUnicodeTextBox





Advanced Tasks

John Harvey
Senior Software Engineer

Advanced Tasks

- Filtering text input
- Letting MLTE handle your Edit menu
- Providing your own CG context
- Customizing drag behavior
- Supporting advanced typography



How Do I Filter Text Input?

- Write your Text Input Carbon event handler
- Install it on the window or control event target
 - Install them *AFTER* object is created
 - Return `evtNotHandledErr` for inactive MLTE objects



Write Your Text Input Event Handler

```
static pascal OSStatus MyTextHandler( EventHandlerCallRef    inHandlerRef,  
                                      EventRef              inEvent,  
                                      void *               inUserData ) {  
  
    // Implement filter code here  
    if ( MyTextFilter( inEvent, inUserData ) == true )  
        return noErr;          // Tell Carbon Event Manager not to pass event on  
    else  
        return evtNotHandledErr; // Pass event on to MLTE
```



Install Handler on the Event Target

```
{  
    const EventTypeSpec specs[] = {  
        { kEventClassTextInput, kEventTextInputUnicodeForKeyEvent } };  
  
    InstallWindowEventHandler( myWindow,  
                             NewEventHandlerUPP( MyTextHandler ),  
                             GetEventTypeCount( specs ), specs,  
                             &myUserData, &myHandlerRef );  
}
```



Can MLTE Maintain My Edit Menu?

- Pass menu and window Carbon event targets to MLTE
 - Make sure Edit menu items have command IDs



Pass Menu and Window Carbon Event Targets to MLTE

```
{  
    // Initialize the TXNCarbonEventInfo data structure  
    TXNCarbonEventInfo    info = { true, 0, kTXNNoAppleEventHandlersMask, NULL };  
    TXNControlTag         tags[] = { kTXNUseCarbonEvents };  
    TXNControlData        data[] = { (UInt32) &info };  
    CFStringRef           keys[] = {  
                                kTXNCommandUpdateKey,  
                                kTXNCommandTargetKey };  
    EventTargetRef        values[] = {  
                                GetMenuEventTarget( myEditMenu ),  
                                GetWindowEventTarget( myWindow ) };  
  
    info.fDictionary = CFDictionaryCreate(  
                                kCFAllocatorDefault, (const void **) &keys,  
                                (const void **) &values, 2,  
                                &kCFCopyStringDictionaryKeyCallbacks,  
                                NULL );  
  
    // Pass the targets to MLTE  
    TXNSetTXNObjectControls( myMLTEObject, false /*clearAllControls*/, 1, tags, data );  
    CFRelease( info.fDictionary );  
}
```



How Do I Pass in My Own CGContextRef?

- Pass the context to **TXNSetTXNObjectControls**
 - Remember to pass in a CG context each time you move the object to a new window via **TXNAttachObjectToWindow**

```
{  
    TXNControlTag    tags[] = { kTXNUseCGContextRefMask };  
    TXNControlData   data[] = { (UInt32) myCGContextRef };  
  
    TXNSetTXNObjectControls( myMLTEObject, false, 1, tags, data );  
    CFRelease( info.fDictionary );  
}
```



How Do I Get Custom Drag Behavior?

- Tell MLTE not to install default drag handlers
- Write your drag tracking and receive handlers
- Install the handlers on the object's window



Tell MLTE Not to Install Its Default Drag Handlers

```
{  
    TXNObject          myMLTEObject;  
    TXNFrameID        myFrameID;  
  
    // Add this mask to your other options  
    TXNFrameOptions    options = kTXNDoNotInstallDragProcsMask;  
  
    TXNNewObject( ..., options, ..., &myMLTEObject, &myFrameID, NULL /*refcon*/ );  
}
```



Write Your Drag Tracking Handler

```
OSErr MyTrackingHandler ( DragTrackingMessage message,
                          WindowPtr theWindow,
                          void * theRefCon,
                          DragReference theDragRef ) {

    // Get your custom drag info
    MyDragInfoType * dragInfo = (MyDragInfoType*) theRefCon;

    // Implement custom tracking behavior here
    MyHandleTracking( message, dragInfo );

    // Let MLTE complete drag tracking operation
    return TXNDragTracker( dragInfo->mlteObject, dragInfo->mlteFrameID,
                           message, theWindow, theDragRef, false );
}
```



Write Your Drag Receive Handler

```
OSErr MyReceiveHandler (      WindowPtr      theWindow,  
                             void *            theRefCon,  
                             DragReference     theDragRef ) {  
  
    MyDragInfoType *   dragInfo = (MyDragInfoType*) theRefCon;  
  
    // Implement your code for filtering dragged data here  
    if ( MyDragDatalsANumber( theDragRef, theRefCon ) )  
    {  
        // Let MLTE complete handling the drag operation  
        return TXNDragReceiver(      dragInfo->mlteObject, dragInfo->mlteFrameID,  
                                   message, theWindow, theDragRef, false );  
    }  
    else  
    {  
        return noErr; // Stops event processing  
    }  
}
```



Install Your Own Drag Tracking and Receive Handlers

```
{  
    MyDragInfoType * dragInfo = new( myMLTEObject, myMLTEFrameID );  
  
    InstallTrackingHandler( NewDragTrackingHandlerUPP( MyTrackingHandler ),  
                           myWindow, (void*) dragInfo /*refcon*/ );  
  
    InstallReceiveHandler( NewDragReceiveHandlerUPP( MyReceiveHandler ),  
                           myWindow, (void*) dragInfo /*refcon*/ );  
}
```

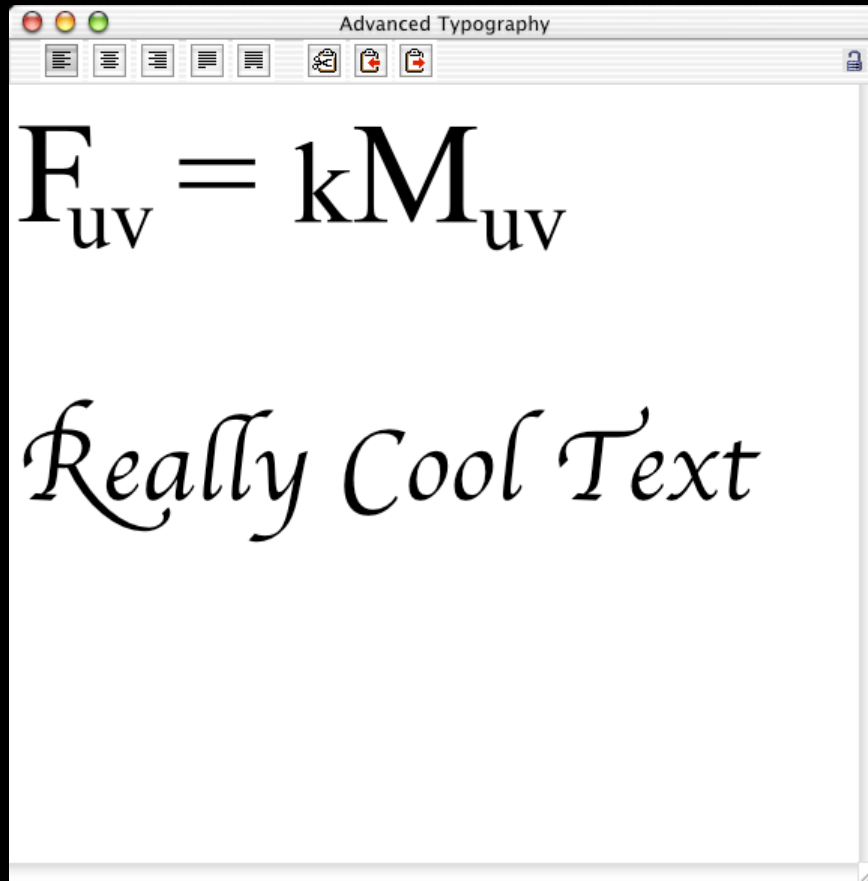


Advanced Typography With ATSUI

- MLTE supports passing in ATSUI style attributes
 - ATSUI provides many more styles than QuickDraw
- Set like any other MLTE style attribute



Examples of ATSUI Attributes



kATSUBeforeWithStreamShiftTag
kATSCrossStreamShiftTag



How Do I Support Advanced Typography?

- Set ATSUI font features and variations on text ranges like other style attributes (e.g., font, color, etc.)

```
{  
    // Use ATSUI APIs to get myFeatureType and myFeatureSelector from font  
  
    TXNATSUIFeatures    features = { 1 /*count*/, &myFeatureType, &myFeatureSelector };  
    TXNTypeAttributes  attributes[] =  
        { { kTXNATSUIFontFeaturesAttribute, sizeof( TXNATSUIFeatures ), { (void*) &features } } };  
  
    TXNSetTypeAttributes( myObject, 1 /*attributeCount*/, attributes,  
        kTXNUseCurrentSelection /*startOffset*/,  
        kTXNUseCurrentSelection /*endOffset*/ );  
}
```



Today's Presentation

- MLTE overview
- MLTE concepts
- Basic tasks
- Advanced tasks
- New tasks—What can you do with Jaguar



New Tasks: Coming With Jaguar

- Attaching a URL to a range of data
- Using your own scrollbars
- Creating monostyled object
- Clearing the undo and redo stack
- Supporting mouse scroll wheel



How Do I Attach a URL to a Range of Text?

- Simply pass the URL as an attribute for a range of data

```
{
    TXNTypeAttributes attributes[] =
        { { kTXNURLAttribute, kTXNURLAttributesSize, { (void*) myURL } } };

    TXNSetTypeAttributes( myObject, 1, attributes,
        kTXNUseCurrentSelection /*startOffset*/,
        kTXNUseCurrentSelection /*endOffset*/ );
}
```



How Can I Use My Own Scrollbars?

- Write your `TrackActionProc` and `LiveTrackingProc`
- Create your own controls
- Write your own scroll callback
- Register the callback with the MLTE object



TrackActionProc

```
void MyTrackActionProcedure ( ControlRef    controlRef,  
                             ControlPartCode  partCode ) {  
  
    // Get myMLTEObject & orientation from custom control property  
  
    // Get units and delta from partCode  
    switch ( partCode ) {  
        case kControlDownButtonPart:  
            units = kTXNScrollUnitsInLines;  
            delta = 1;  
            break;  
    }  
  
    // Have MLTE scroll the object  
    if ( orientation == kTXNHorizontal )  
        TXNScroll( myMLTEObject, 0 /*vUnit*/, units /*hUnit*/, 0 /*vDelta*/, delta /*hDelta*/ );  
    else  
        TXNScroll( myMLTEObject, units /*vUnit*/, 0 /*hUnit*/, delta /*vDelta*/, 0 /*hDelta*/ );  
}
```



Write Your Own Scroll Callback

```
void MyScrollInfoProc (          SInt32          iValue,  
                           SInt32          iMaxValue,  
                           TXNScrollBarOrientation iScrollBarOrientation,  
                           SInt32          iUserData ) {  
  
    MyScrollInfoData * scrollInfo = (MyScrollInfoData*) iUserData;  
  
    // Get scrollbar to be updated  
    if( iScrollBarOrientation == kTXNVertical )  
        controlRef = scrollInfo->vScrollbar;  
    else  
        controlRef = scrollInfo->hScrollbar;  
  
    // Update the control  
    MyUpdateScrollValues( controlRef, iValue, iMaxValue );  
}
```



Register the Callback

```
{  
    MyScrollInfoData      scrollInfo;  
    TXNScrollInfoUPP     scrollInfoUPP;  
  
    // Store the information needed by the callback  
    scrollInfo.vScrollBar = myVScrollbarRef;  
    scrollInfo.hScrollBar = myHScrollbarRef;  
  
    scrollInfoUPP = NewTXNScrollInfoUPP( MyScrollInfoProc );  
  
    TXNRegisterScrollInfoProc( myMLTEObject, scrollInfoUPP, (SInt32) &scrollInfo /*userData*/ );  
}
```



How Do I Force a Single Style in My Object?

```
{
    TXNObject          myMLTEObject;
    TXNFrameID        myFrameID;

    // Add this mask to your other options
    TXNFrameOptions    options = kTXNMonostyledTextMask;

    TXNNewObject ( ..., options, ..., &myMLTEObject, &myFrameID, NULL /*refcon*/ );
}
```



How Do I Get Support for the Mouse Scroll Wheel?

- Do nothing!





Demo

New Features

Many New MLTE Features

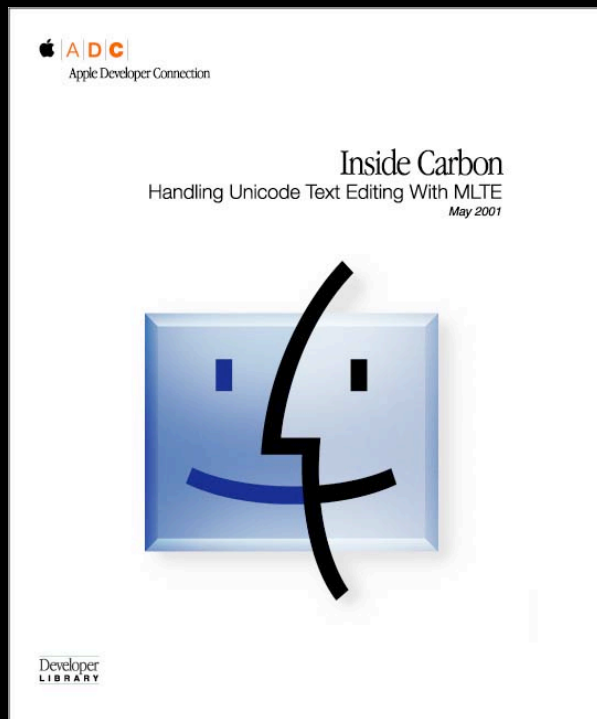
- URL support
- Monostyled text
- Support for scrolling with controls created by client
- Clear undo and redo stack
- Support for mouse scroll wheel

Try them out!



Documentation

Multilingual Text Engine



- Handling Unicode Text With MLTE
- Multilingual Text Engine Reference
- Setting Up MLTE to Use Carbon Events

Documentation > Carbon > Text and Other International > MLTE

developer.apple.com/techpubs/macosx/Carbon/text/MultilingualTextEngine/multilingualltextengine.html



More Documentation

Multilingual Text Engine

- TN 2026: Multilingual Text Engine
developer.apple.com/technotes/indexes/te-a.html
- Header file comments (Jaguar APIs)
Frequently Asked Questions
[/System/Library/Frameworks/Carbon.framework/
Frameworks/ HIToolbox.framework/Headers](#)
- Sample code
developer.apple.com/sdk/index.html/

Roadmap

201 Font Manager

Room C
Tue., 10:30am

202 Drawing Text With ATSUI

Room J
Tue., 3:30pm

**206 HIToolbox: New Controls
and Services**

Hall 2
Wed., 2:00pm

FF008 International Technologies

Room J1
Thurs., 5:00pm



Who to Contact

Xavier Legros

Mac OS X Evangelist

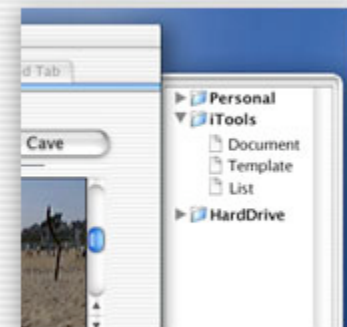
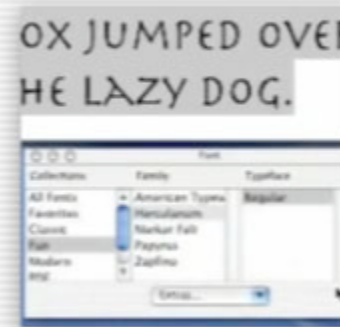
xavier@apple.com

<http://developer.apple.com/wwdc2002/urls.html>





Q&A



Xavier Legros
Mac OS X Evangelist
xavier@apple.com

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 **WWDC2002**